**Mockito Hands-On Exercises**

**Exercise 1: Mocking and Stubbing**

**ExternalApi.java:**

package com.example;

public interface ExternalApi {

String getData();

}

**MyService.java:**

package com.example;

public class MyService {

private ExternalApiapi;

public MyService(ExternalApiapi) {

this.api = api;

}

public String fetchData() {

return api.getData();

}

}

**MyServiceTest.java:**

package com.example;

import static org.mockito.Mockito.\*;

import static org.junit.Assert.\*;

import org.junit.Test;

import org.mockito.Mockito;

public class MyServiceTest {

*@Test*

public void testExternalApi() {

ExternalApimockApi = Mockito.*mock*(ExternalApi.class);

*when*(mockApi.getData()).thenReturn("Mock Data");

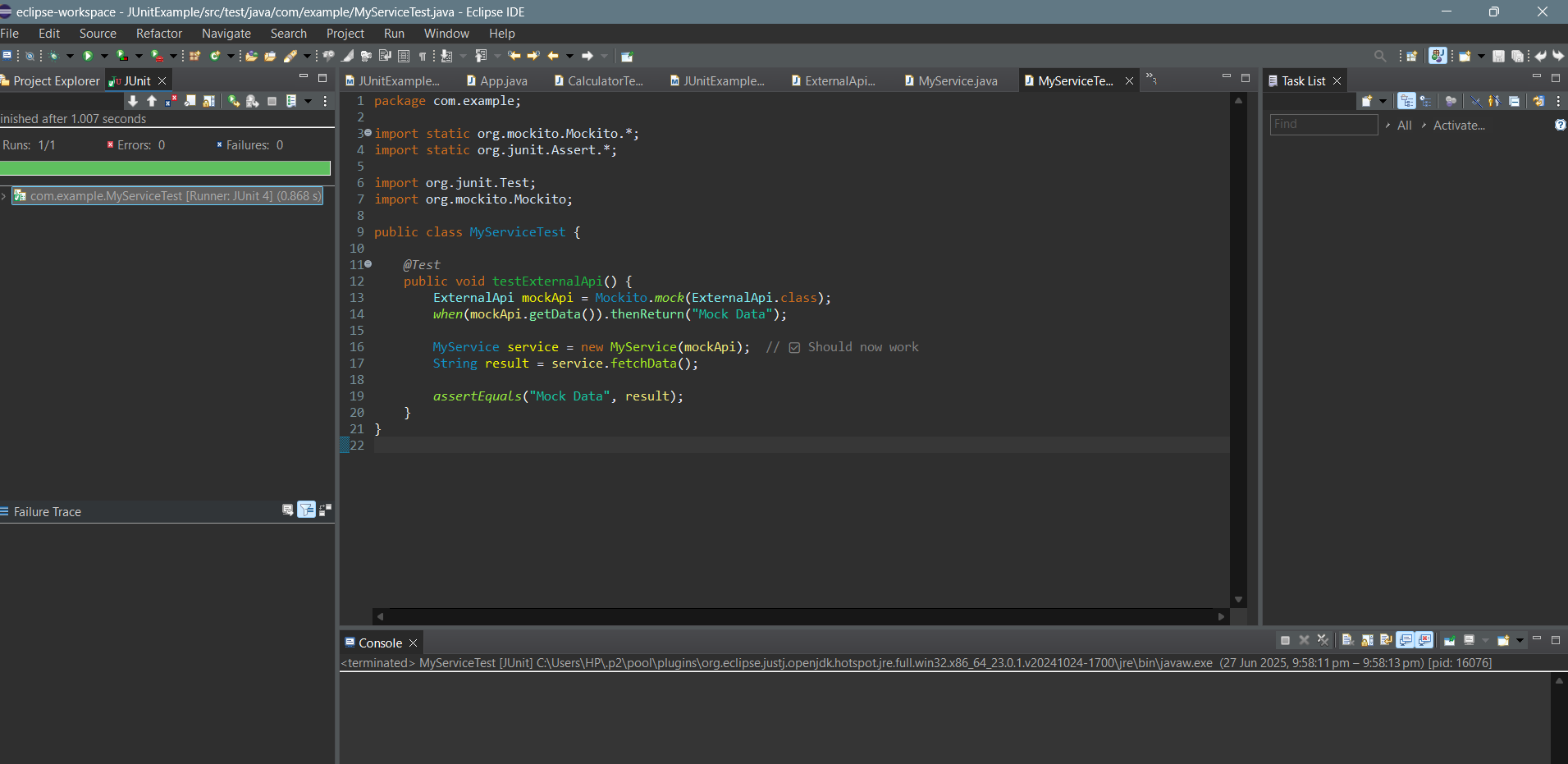
MyService service = new MyService(mockApi); // ✅ Should now work

String result = service.fetchData();

*assertEquals*("Mock Data", result);

}

}

 **Exercise 2: Verifying Interactions**

**ExternalApi.java:**

package com.example;

public interface ExternalApi {

String getData();

}

**MyService.java**:

package com.example;

public class MyService {

private ExternalApiapi;

public MyService(ExternalApiapi) {

this.api = api;

}

public String fetchData() {

return api.getData(); // This should be verified in test

}

}

**MyServiceTest.java:**

package com.example;

import static org.mockito.Mockito.\*;

import org.junit.Test;

import org.mockito.Mockito;

public class MyServiceTest {

@Test

public void testVerifyInteraction() {

ExternalApimockApi = Mockito.mock(ExternalApi.class);

MyService service = new MyService(mockApi);

service.fetchData();

verify(mockApi).getData();

}

}

